A MATLAB PROTOTYPING PLATFORM FOR REAL-SPACE DENSITY FUNCTIONAL THEORY

A Thesis
Presented to
The Academic Faculty

By

Qimen Xu

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Civil and Environmental Engineering

Georgia Institute of Technology

December 2020

A MATLAB PROTOTYPING PLATFORM FOR REAL-SPACE DENSITY FUNCTIONAL THEORY

Thesis committee:

Dr. Phanish Suryanarayana
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Andrew Medford
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

Date approved: December 4, 2020

Dedicated to my parents, my grandparents, and my uncle,

for their constant support.

ACKNOWLEDGMENTS

I would like to thank Professor Phanish Suryanarayana, my advisor, for his patient guidance and help throughout this work. I would also like to thank the members of my thesis committee, Professor Edmond Chow and Professor Andrew Medford, for serving on my thesis committee and giving invaluable feedback. I thank Dr. John Pask, for his insightful suggestions and help with my research.

I thank all my friends here at Georgia Tech. Special thanks to Abhiraj Sharma, my friend and colleague, who made great contributions to the preparation of this work. Swarnava Gosh, and Phanisri (Pradeep) Pratapa, for their kind help when I got started in the group. I would like to thank Mostafa Shojaei, Ajinkya Lokhande, Ajit Kamath, Shashikant Kumar, Arpit Bhardwaj, Boqin Zhang, Xin Jing, Hua Huang, Benjamin Comer, and Xiangyun (Ray) Lei, I enjoyed invaluable conversations and discussions with them.

I gratefully acknowledge the support for this work by the U.S. Department of Energy, Office of Science, under the grant DE-SC0019410. Initial development efforts were supported by the grants 1333500 and 1553212 funded by the U.S. National Science Foundation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Over the past few decades, Kohn-Sham Density Functional Theory (DFT) has become one of the most widely used electronic structure theories for understanding and predicting materials properties from the first principles of quantum mechanics, without the need for any empirical or adjustable parameters. Despite its high accuracy to cost ratio compared to other such ab initio methods, the efficient solution of the Kohn-Sham problem remains a formidable task. Therefore, nearly all established DFT codes take advantage of the efficiency and parallel scalability provided by lower level programming languages such as C, C++, and Fortran. However, the effort to achieve better parallel scalability in these lower level programming languages inevitably leads to significant code complexity, making the testing of new ideas and methods prone to error, while requiring excessively large effort for implementation. As a consequence, the rate of advancements in the field can be significantly hindered.

In this thesis we present M-SPARC: MATLAB-Simulation Package for Ab-initio Real-space Calculations. It can perform pseudopotential spin-polarized and unpolarized Kohn-Sham Density Functional Theory (DFT) simulations for isolated systems such as molecules as well as extended systems such as crystals, surfaces, and nanowires. M-SPARC provides a rapid prototyping platform for the development and testing of new algorithms and methods in real-space DFT, with the potential to significantly accelerate the rate of advancements in the field. It also provides a convenient avenue for the accurate first principles study of small to moderate sized systems.

CHAPTER 1
INTRODUCTION

Kohn-Sham Density Functional Theory (DFT) [1, 2] is one of the most widely used electronic structure theories for understanding and predicting materials properties from the first principles of quantum mechanics [3, 4], without the need for any empirical or adjustable parameters. The popularity of DFT can be attributed to its high accuracy to cost ratio when compared to other such ab initio methods. However, the efficient solution of the Kohn-Sham problem remains a significant challenge, which limits the size and length scales that are accessible to such a rigorous first principles investigation.

The computational cost and memory associated with diagonalization based solutions of the Kohn-Sham equations scale cubically and quadratically with respect to system size, respectively [5]. The accompanying prefactors are particularly large when systematically improvable discretizations are used. Though the Kohn-Sham problem can be reformulated in terms of the truncated density matrix to develop linear scaling methods [6, 7], they suffer from a number of limitations. These include being restrictive in the types of systems that can be studied and having a significantly larger prefactor, which makes them inefficient relative to their cubic scaling counterparts for small to moderate system sizes [8, 9].

In view of the above bottlenecks, nearly all established DFT codes take advantage of the efficiency and parallel scalability provided by lower-level programming languages such as C, C++, and Fortran. In particular, it is common to employ multiple levels of parallelization, the nature of which also changes between different parts of the code. This leads to significant code complexity, making the testing of new ideas and methods prone to error while requiring excessively large effort for implementation. As a consequence, the rate of advancements in the field can be significantly hindered. This motivates the development of simple but accurate codes written in high-level programming languages such as Python

and MATLAB, which enable rapid prototyping.

KSSOLV [10] is one such MATLAB code for the plane-wave method, traditionally the discretization of choice in Kohn-Sham DFT [11, 12, 13, 14, 15, 16]. However, to the best of our knowledge, no such counterpart exists for real-space methods, which have gained significant attention recently [17, 18, 19, 20, 21, 22, 23, 24], in part due to their high scalability for large-scale parallel computing [25, 23, 24], flexibility in the choice of boundary conditions [26, 27, 28], and amenability to the development of linear scaling methods [29, 8]. Motivated by this, in this work we develop M-SPARC: MATLAB-Simulation Package for Ab-initio Real-space Calculations. It provides a rapid prototyping platform for the development and testing of new algorithms and methods in real-space DFT. Additionally, it provides a convenient avenue for the accurate first principles study of small to moderate sized systems.

The remainder of this thesis is organized as follows. In Chapter 2 we introduce the formulation of real-space DFT implemented in M-SPARC, followed by an overview of the M-SPARC software architecture in Chapter 3. In Chapter 4, we demonstrate the major functions through illustrative examples. Next, we describe the impact of this work in Chapter 5. Finally, we provide concluding remarks in Chapter 6.

CHAPTER 2

REAL-SPACE DFT FORMULATION

In this chapter, we introduce the real-space DFT formulation implemented in M-SPARC. For the sake of simplicity, we will focus on the formulation of spin-unpolarized systems, though spin-polarized systems are also supported in M-SPARC. Consider a system with unit cell Ω, which contains N_A atoms with a total of N_e valence electrons. Denote the collection of atomic positions by $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{N_A}\}$. The central focus of Kohn-Sham DFT is the solution of the Kohn-Sham equations [2]:

$$\left(\mathcal{H} \equiv -\frac{1}{2}\nabla^2 + V_{\text{eff}}\left(\rho; \mathbf{R}\right)\right)\psi_n = \lambda_n \psi_n\,, \quad n = 1, 2, \ldots, N_s\,, \tag{2.1}$$

where $\mathcal{H}$ is the Hamiltonian, ∇^2 is the Laplace operator, ψ_n are the Kohn-Sham orbitals with energies λ_n, V_{eff} is the effective potential, N_s is the number of states. In addition, ρ is the electron density, which can be written as

$$\rho(\mathbf{x}) = 2\sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) |\psi_n(\mathbf{x}, \mathbf{k})|^2 \, \mathrm{d}\mathbf{k}\,, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3\,, \tag{2.2}$$

where g_n are the orbital occupations, typically given by the Fermi-Dirac distribution,

$$g_n(\mathbf{k}) = \left(1 + \exp\left(\frac{\lambda_n(\mathbf{k}) - \lambda_f}{k_B T}\right)\right)^{-1}\,,$$

or the Gaussian smearing function,

$$g_n(\mathbf{k}) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{\lambda_n(\mathbf{k}) - \lambda_f}{k_B T}\right)\right)\,,$$

where k_B is the Boltzmann constant, T is the temperature and λ_f is the Fermi energy, which is chosen such that

$$2\sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k})\, \mathrm{d}\mathbf{k} = N_e\,.$$

In computations, the boundary conditions applied on the Kohn-Sham orbitals $\psi_n(\mathbf{x})$ depend on whether the system is finite or periodic in a certain direction. If the system is finite in one direction, zero Dirichlet boundary condition is applied, i.e.,

$$\psi_n(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_D\,, \quad n = 1, 2, \ldots, N_s\,, \tag{2.3}$$

where Γ_D denotes the domain boundary in the finite directions. On the other hand, if the system is periodic in a certain direction, we apply Bloch-periodic boundary conditions, i.e., for lattice vector $\mathbf{L}$ and Bloch wavevector $\mathbf{k}$,

$$\psi_n(\mathbf{x} + \mathbf{L}, \mathbf{k}) = e^{i\mathbf{k}\cdot\mathbf{L}}\psi_n(\mathbf{x}, \mathbf{k}), \quad \mathbf{x} \in \Omega\,, \quad n = 1, 2, \ldots, N_s\,. \tag{2.4}$$

The effective potential term V_{eff} in the Hamiltonian $\mathcal{H}$ consists of three components,

$$V_{\text{eff}} = \phi + V_{\text{xc}} + V_{\text{nl}}\,,$$

where ϕ is the electrostatic potential, V_{xc} is the exchange correlation potential, and V_{nl} is the nonlocal pseudopotential operator. A local formulation of the electrostatics in terms of ionic pseudocharges is employed [30, 31], wherein the electrostatic potential ϕ is obtained by the solution of the Poisson equation:

$$-\frac{1}{4\pi}\nabla^2\phi = \rho(\mathbf{x}) + b(\mathbf{x}; \mathbf{R})\,, \tag{2.5}$$

where b is the pseudocharge density. Dirichlet and periodic boundary conditions are prescribed on the electrostatic potential along directions in which the system is finite and

extended, respectively. The Dirichlet boundary condition values are determined using a multipole expansion for isolated systems and a dipole correction for surfaces and nanowires [32, 26]. The nonlocal pseudopotential operator V_{nl} takes the form [33]

$$V_{\text{nl}}\psi_n = \sum_{J=1}^{N_A} \sum_{lm} \gamma_{Jl} \tilde{\chi}_{Jlm} \int_{\Omega} \tilde{\chi}_{Jlm}(\mathbf{x}, \mathbf{k}; \mathbf{R}_J) \psi_n(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,, \tag{2.6}$$

where $\tilde{\chi}_{Jlm}$ are the sum of the Bloch-periodically mapped nonlocal projection functions associated with atom J and its periodic images, γ_{Jl} are the associated coefficients, l and m are the azimuthal and magnetic quantum numbers, respectively.

Note that the Kohn-Sham problem (Equation 2.1) is a nonlinear eigenvalue problem since the Hamiltonian operator $\mathcal{H}$ depends on the electron density ρ, but ρ in turn depends on the eigenfunctions ψ_n as indicated in Equation 2.2. Once the electronic ground state is determined through the solution to the Kohn-Sham problem, the free energy can be obtained by [24]

$$\begin{aligned} \mathcal{F}(\mathbf{R}) &= 2 \sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \lambda_n(\mathbf{k}) \, \mathrm{d}\mathbf{k} - \int_{\Omega} V_{xc}(\rho(\mathbf{x})) \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} + E_{xc}(\rho(\mathbf{x})) \\ &+ \frac{1}{2} \int_{\Omega} (b(\mathbf{x}; \mathbf{R}) - \rho(\mathbf{x})) \phi(\mathbf{x}; \mathbf{R}) \, \mathrm{d}\mathbf{x} + E_{sc}(\mathbf{R}) - TS \,, \end{aligned} \tag{2.7}$$

where E_{xc} is the exchange-correlation energy, E_{sc} accounts for the self energy and the repulsive energy correction associated with the pseudocharge densities (the explicit form can be found in Appendix A). S is the electronic entropy, which, for Fermi-Dirac smearing, takes the form

$$S = -2k_B \sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \log g_n(\mathbf{k}) + (1 - g_n(\mathbf{k})) \log(1 - g_n(\mathbf{k})) \, \mathrm{d}\mathbf{k} \,, \tag{2.8}$$

and for Gaussian smearing,

$$S = k_B \sum_{n=1}^{N_s} \fint_{BZ} \frac{1}{\sqrt{\pi}} \exp\left(- \left(\frac{\lambda_n(\mathbf{k}) - \lambda_f}{k_B T} \right)^2 \right) \mathrm{d}\mathbf{k} \,. \tag{2.9}$$

The atomic force on the J^{th} atom is obtained by [24]

$$\begin{aligned} \mathbf{f}_J &= -\frac{\partial \mathcal{F}(\mathbf{R})}{\partial \mathbf{R}_J} \\ &= \sum_{J'} \int_\Omega \nabla b_{J'}(\mathbf{x}; \mathbf{R}_{J'}) \phi(\mathbf{x}; \mathbf{R}) \, \mathrm{d}\mathbf{x} + \mathbf{f}_{sc,J}(\mathbf{R}) \\ &- 4 \sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \sum_{lm} \gamma_{Jl} \Re \left[\left(\int_\Omega \psi_n^*(\mathbf{x}, \mathbf{k}) \tilde{\chi}_{Jlm}(\mathbf{x}; \mathbf{R}_J) \, \mathrm{d}\mathbf{x} \right) \right. \\ &\quad \left. \left(\times \int_\Omega \nabla \psi_n(\mathbf{x}, \mathbf{k}) \tilde{\chi}^*_{Jlm}(\mathbf{x}; \mathbf{R}_J) \, \mathrm{d}\mathbf{x} \right) \right] \mathrm{d}\mathbf{k} \,, \end{aligned} \tag{2.10}$$

where the summation index J' in the first term runs over all the periodic images of atom J as well as the J^{th} atom itself, $b_{J'}$ is the pseudocharge density of the periodic image J' , $\mathbf{f}_{sc,J} \equiv -\frac{\partial E_{sc}(\mathbf{R})}{\partial \mathbf{R}_J}$ (see Appendix A for the explicit formula), and $\Re[\cdot]$ denotes the real part of the bracketed expression. Apart from the free energy and atomic forces, M-SPARC also implements the calculation of the stress tensor. The stress tensor takes the form [34]

$$\begin{aligned} \sigma_{\alpha\beta} &= \frac{1}{|\Omega|} \Bigg[- 2 \sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \int_\Omega \nabla_\alpha \psi_n^*(\mathbf{x}, \mathbf{k}) \nabla_\beta \psi_n(\mathbf{x}, \mathbf{k}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{k} \\ &+ \delta_{\alpha\beta} \left(E_{xc} - \int_\Omega V_{xc}\big(\rho, \boldsymbol{\nabla}\rho\big) \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right) - \int_\Omega \rho(\mathbf{x}) \frac{\partial \varepsilon_{xc}\big(\rho(\mathbf{x}), \boldsymbol{\nabla}\rho(\mathbf{x})\big)}{\partial\big(\nabla_\beta \rho(\mathbf{x})\big)} \nabla_\alpha \rho(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &- \delta_{\alpha\beta} E_{nl} - 4 \sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \sum_{J=1}^{N_A} \sum_{lm} \gamma_{Jl} \Re \Bigg[\Bigg(\sum_{J'} \int_\Omega \chi^*_{J'lm}(\mathbf{x}; \mathbf{R}_{J'}) \\ &\times e^{i\mathbf{k}\cdot(\mathbf{R}_J - \mathbf{R}_{J'})} \big(\mathbf{x} - \mathbf{R}_{J'}\big)_\beta \nabla_\alpha \psi_n(\mathbf{x}, \mathbf{k}) \, \mathrm{d}\mathbf{x} \Bigg) \Bigg(\int_\Omega \tilde{\chi}_{Jlm}(\mathbf{x}; \mathbf{R}_J, \mathbf{k}) \psi_n^*(\mathbf{x}, \mathbf{k}) \, \mathrm{d}\mathbf{x} \Bigg) \Bigg] \mathrm{d}\mathbf{k} \\ &+ \frac{1}{4\pi} \int_\Omega \nabla_\alpha \phi(\mathbf{x}; \mathbf{R}) \nabla_\beta \phi(\mathbf{x}; \mathbf{R}) \, \mathrm{d}\mathbf{x} + \sum_I \int_\Omega \nabla_\alpha b_I(\mathbf{x}; \mathbf{R}_I) \big(\mathbf{x} - \mathbf{R}_I\big)_\beta \phi(\mathbf{x}; \mathbf{R}) \, \mathrm{d}\mathbf{x} \\ &+ \frac{1}{2} \delta_{\alpha\beta} \int_\Omega \big(b(\mathbf{x}; \mathbf{R}) - \rho(\mathbf{x}) \big) \phi(\mathbf{x}; \mathbf{R}) \, \mathrm{d}\mathbf{x} \Bigg] + \sigma^{E_{sc}}_{\alpha\beta} \,, \end{aligned} \tag{2.11}$$

where $|\Omega|$ denotes the volume of the unit cell Ω, $\nabla_\alpha(\alpha = 1, 2, 3)$ denotes the α^{th} components of the gradient vector, the summation index I runs over all atoms in $\mathbb{R}^3$, J' runs over all the periodic images of the J^{th} atom as well as the J^{th} atom, E_{nl} is the nonlocal pseudopotential energy, which takes the form [33]

$$E_{nl} = 2\sum_{n=1}^{N_s} \fint_{BZ} g_n(\mathbf{k}) \sum_{J=1}^{N_A} \sum_{lm} \gamma_{Jl} \left| \int_\Omega \tilde{\chi}^*_{Jlm}(\mathbf{x}, \mathbf{R}_J, \mathbf{k}) \psi_n(\mathbf{x}, \mathbf{k}) \, \mathrm{d}\mathbf{x} \right|^2 \mathrm{d}\mathbf{k} \,, \tag{2.12}$$

and $\sigma^{E_{sc}}_{\alpha\beta}$ is the pseudocharge correction term, whose explicit form can be found in Appendix A. The reader is referred to [34] for more details regarding the derivation of the real-space formulation of the stress tensor.

CHAPTER 3
SOFTWARE DESCRIPTION

3.1 Software Architecture

M-SPARC is written exclusively using the MATLAB language. It employs a high-order central finite-difference approximation for discretization of the equations and a trapezoidal rule for spatial integrations. A pictorial overview of the M-SPARC framework for performing DFT calculations is illustrated in Figure 3.1. It requires two input files: (i) `.inpt` file containing user options and parameters, including the dimensions of the cell, boundary conditions, information about the finite-difference grid, and choice of exchange-correlation functional; and (ii) `.ion` file containing the atomic information, including the atom type, its spatial position, and path to its pseudopotential file.

Three types of calculations can be performed in M-SPARC: single point, structural relaxation, and molecular dynamics. In single point calculations, the electronic ground-state is calculated for a given atomic configuration. In structural relaxation simulations, the energy is minimized with respect to the atomic positions or volume of the cell, while employing the computed Hellmann-Feynman atomic forces [23, 24] or stress tensor [34], respectively. In molecular dynamics simulations, the ionic positions, velocities, and accelerations are evolved by integrating the equations of motion, while utilizing the atomic forces. Indeed, the electronic ground state needs to be determined for every atomic configuration encountered during the structural relaxation and molecular dynamics simulations.

The electronic ground state is determined using the self-consistent field (SCF) method [5]. Specifically, a fixed-point iteration is performed with respect to either the electron density or the potential. For the very first electronic ground state calculation, the superposition of isolated atom electron densities is used as initial guess for the electron density, whereas

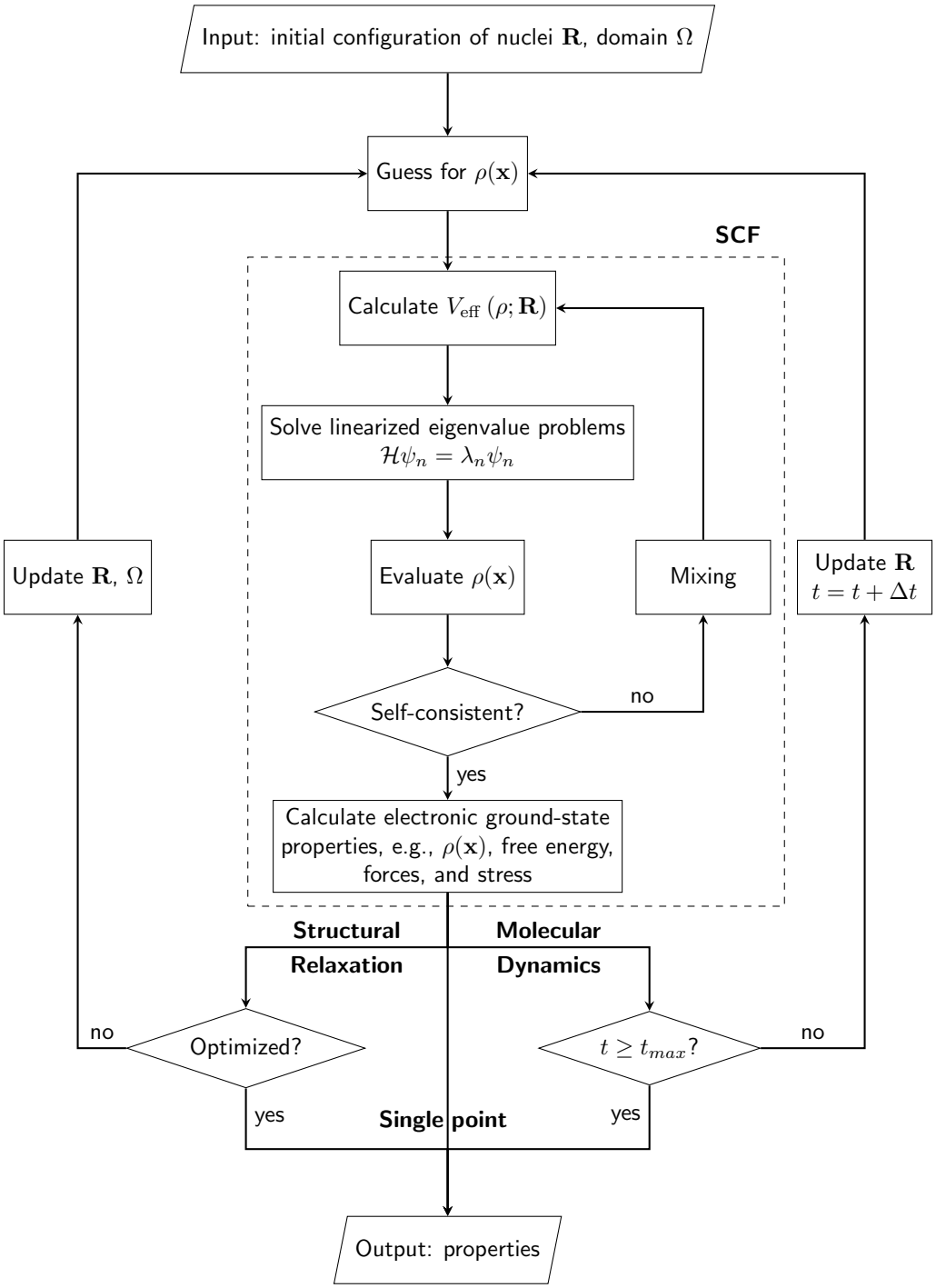

Figure 3.1: Overview of the M-SPARC framework for performing DFT calculations.

for every subsequent such calculation, extrapolation based on previous solutions is used [35]. The convergence of the SCF iteration is accelerated using the restarted variant of the Periodic Pulay mixing scheme [36, 37], with the option of real-space preconditioning

[38]. For spin-polarized calculations, mixing is performed simultaneously for both spin components, i.e., using a vector of twice the original length containing both spin-up and spin-down density/potential components. The SCF iteration is considered to be converged when self-consistency in the solution is achieved.

In each SCF iteration, partial diagonalization of the linearized eigenproblem is performed using the CheFSI method [39, 40], with multiple Chebyshev filtering steps in the very first iteration of the entire DFT simulation [41]. Zero Dirichlet (Equation 2.3) and Bloch-periodic boundary conditions (Equation 2.4) are prescribed on the orbitals ψ_n along the directions in which the system is finite and extended, respectively. While performing the Hamiltonian-matrix products, the Kronecker product formulation for the Laplacian [42] is used, with the remaining terms handled in a matrix-free fashion. A local formulation of the electrostatics in terms of ionic pseudocharges is employed [30, 31], wherein the electrostatic potential ϕ (component of V_{eff}) is determined by the solution of the Poisson equation (Equation 2.5). Dirichlet and periodic boundary conditions are prescribed on the electrostatic potential along directions in which the system is finite and extended, respectively. The Dirichlet boundary condition values are determined using a multipole expansion for isolated systems and a dipole correction for surfaces and nanowires [32, 26]. The linear system is solved using the AAR method [43, 44] in conjunction with incomplete Cholesky preconditioning.

After completion of the DFT simulation, in addition to the parameters used in the calculation, all quantities of interest such as the orbitals, occupations, electron density, and electrostatic potential are stored in the structure denoted by `S`. General information such as input parameters, progress of the SCF iteration, energy, maximum force, and timing are written into the `.out` file. Additionally, `.static`, `.geopt`, `.cellopt`, and `.aimd` files are written for single point, atomic structural relaxation, cell structural relaxation, and molecular dynamics simulations, respectively. The `.static` file contains the atomic positions and forces; the `.geopt` file contains the atomic positions and forces for each atomic

relaxation step; the `.cellopt` contains the cell information and stress tensor for each cell relaxation step; and the `.aimd` file contains atomic positions, forces, and velocities. Note that a `.restart` file is also written for structural relaxations and molecular dynamics, which can be used to restart the simulation.

3.2 Software Functionalities

M-SPARC can perform spin-polarized and unpolarized ab initio calculations based on pseudopotential Kohn-Sham DFT for isolated systems such as molecules as well as extended systems such as crystals, surfaces, and nanowires. Specifically, it can currently perform single point calculations for a given atomic configuration, structural relaxations with respect to atomic positions or cell volume, and NVE molecular dynamics simulations, all using either the ONCV [45] or Troullier-Martin pseudopotentials [46] and either the LDA [47, 48] or GGA [49] exchange-correlation functionals. In so doing, M-SPARC can calculate the energy of the system as well as the Hellmann-Feynman atomic forces and stress tensor. The output from such DFT calculations can be used to calculate a number of properties, such as equilibrium bond lengths, HOMO-LUMO gap, dipole moment, surface energy, cohesive energy, defect energy, lattice constant, elastic moduli, density of states, electronic band structure, and pair correlation function.

3.3 Open Source Release

The source code of M-SPARC is released under the GPL-3.0 License and it can be downloaded from the following GitHub page: https://github.com/SPARC-X/M-SPARC. A screenshot of the webpage is shown in Figure 3.2. Interested users and developers are welcome to use and contribute to the code.

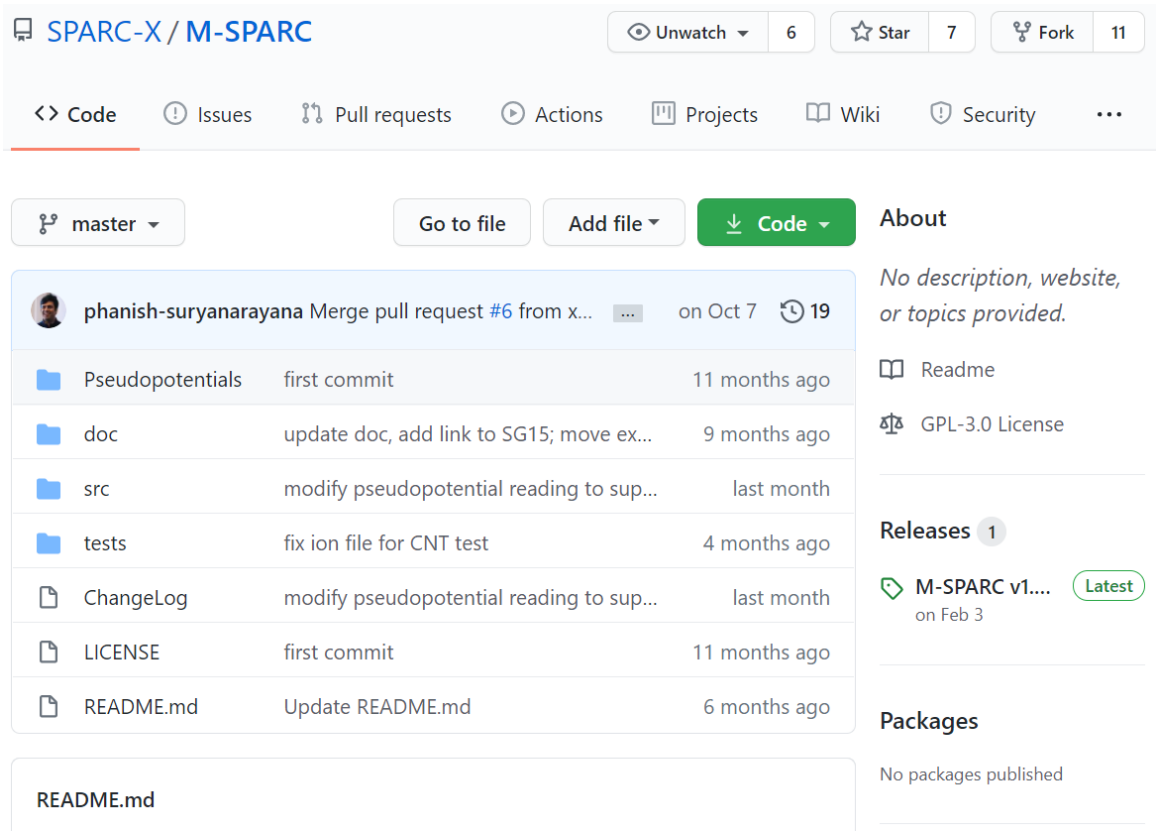

Figure 3.2: A screenshot of the M-SPARC release GitHub page.

3.3.1 A tiny example

Here we provide an example to show the usage of M-SPARC. Once the user downloads the source code successfully, launch MATLAB and change directory to the `M-SPARC/src` folder. The main function is written in the source file named `msparc.m`.

In this example, we will use the input files located in `M-SPARC/tests/examples/`. On the MATLAB command window, run the following commands to launch the simulation

```
fname = '../tests/examples/BaTiO3';
S = msparc(fname);
```

Make sure the current folder is the `src/` folder, otherwise MATLAB will raise an error stating

```
'msparc' is not found in the current folder or on the MATLAB path.
```

One workaround is to add the `src/` folder to the MATLAB path using the `addpath` command.

Once the simulation is done, a `BaTiO3.out` file and a `BaTiO3.static` file will be generated in the `M-SPARC/tests/examples/` folder, i.e., where the input files are located. The output files contain general information such as the input parameters and output results like free energy and atomic forces. Moreover, the returned structure `S` contains most quantities of interest, such as the Kohn-Sham orbitals ψ_n (`S.psi`), eigenvalues λ_n (`S.EigVal`), occupations g_n (`S.occ`), pseudocharge density b (`S.b`), electron density ρ (`S.rho`), and electrostatic potential ϕ (`S.phi`), just to list a few, which are useful for debugging and post-processing directly within MATLAB.

3.3.2 SPARC

SPARC [50] is a high-performance version of the M-SPARC code written in the `C` language, which achieves efficient parallelism through the message passing interface (`MPI`) [51]. The source code of SPARC is released under the GPL-3.0 License and can be downloaded from the following GitHub page: https://github.com/SPARC-X/SPARC. It is straightforward to install/use and highly competitive with state-of-the-art plane-wave codes, especially on large numbers of processes.

M-SPARC and SPARC share the same structure, algorithms, input, and output files, which makes M-SPARC the perfect prototyping platform for new algorithms and methods before their parallel implementations in SPARC. Examples include the development of the real-space formulation of the Hellmann-Feynman stress tensor in DFT [34], and the real-space formulation for isotropic Fourier-space preconditioners that can accelerate the SCF iteration in DFT [38], which are now implemented in both codes. In fact, M-SPARC has played a vital role during the implementation of SPARC. Not only does it provide a clean structure of the entire program, but it also provides key debugging references at every stage in the implementation of SPARC.

Apart from the fact that SPARC achieves parallelism via `MPI` in `C`, while M-SPARC is a serial code written in the high-level MATLAB language, there are a few key differences that are worth noting when converting an M-SPARC implementation to a SPARC implementation. First, while reading input from an input file in M-SPARC is straightforward, SPARC reads the input file by the root process first and then broadcasts the information to all processes since `MPI` processes do not share data. Second, while performing the Laplacian-matrix/vector products, M-SPARC implements the Kronecker product formulation [42], which also takes advantage of the sparse matrix-matrix routines available in MATLAB. On the other hand, SPARC implements a matrix-free fashion. Third, the solution to the Kohn-Sham equations (Equation 2.1) is achieved through iterative methods, wherein the Kohn-Sham orbitals ψ_n are randomly initialized. Since the initial orbitals are random, the extent to which the answers between M-SPARC and SPARC match is dependent on the tolerances. For the purpose of debugging, sometimes one can manually force the initial orbitals to be the same in both M-SPARC and SPARC, to remove the randomness introduced by the initial orbitals. Fourth, during the CheFSI method [39, 40], a subspace eigenvalue problem has to be solved in each SCF iteration. While M-SPARC solves the standard eigenvalue problem, which is the procedure proposed in the original method, SPARC solves a generalized eigenvalue problem, to avoid the need to orthogonalize the orbitals before the Rayleigh-Ritz step. However, this should not change the result since they are mathematically equivalent. Last but not least, since all data are destroyed after a `C` program is completed, output data from SPARC are stored through output files, whereas M-SPARC keeps all the data in a structure and returns it as an output, in addition to the common output files. For example, the electron density is written in a `.dens` file in SPARC, while M-SPARC saves the electron density in the output structure `S` and the electron density can be accessed by `S.rho`.

CHAPTER 4
ILLUSTRATIVE EXAMPLES

In this chapter, we demonstrate some of the major functionalities of M-SPARC through representative examples. Input and output files for these examples are available as part of the distribution, which can be found in the `M-SPARC/tests/IllustrativeExamples/` folder. We employ ONCV pseudopotentials [45] with 19, 14, 4, 4, and 1 valence electrons for gold, germanium, carbon, silicon, and hydrogen, respectively. In all cases, we use the 12^{th} order finite-difference approximation for the discretization of the equations. The results are compared to the highly converged results obtained by the established plane-wave code ABINIT [14]. The simulations are performed on a single `hive-himem` computer node on the `Hive` cluster at Georgia Institute of Technology, with dual-socketed Intel Xeon Gold 6226 (Cascade Lake) 2.7 GHz 12-core CPUs, 1.5 TB of DDR4-2933 ECC RAM, Mellanox EDR (100 Gb/s) Infiniband networking, and Red Hat Enterprise Linux 7.6. The detailed timing of each test can be found in the corresponding output files available in the distribution.

4.1 Bulk fcc gold

In this section, we demonstrate the ability of M-SPARC to accurately calculate the lattice constant of a bulk system. We choose the single 4-atom unit cell bulk fcc gold system with LDA, $7 \times 7 \times 7$ grid for Brillouin zone integration, and mesh-size of 0.3 Bohr. The system is illustrated in Figure 4.1. We compare the results obtained from ABINIT, where a plane-wave cutoff of 30 Ha is used. In Figure 4.2, we plot the computed energy and pressure as functions of the lattice constant as well as their curve fits using cubic splines. In Table 4.1, we show the lattice constants and cohesive energies obtained in M-SPARC and ABINIT. We observe that the results obtained from M-SPARC and ABINIT are in good agreement.

Each simulation with a single lattice constant takes around $13 - 21$ minutes in M-SPARC.

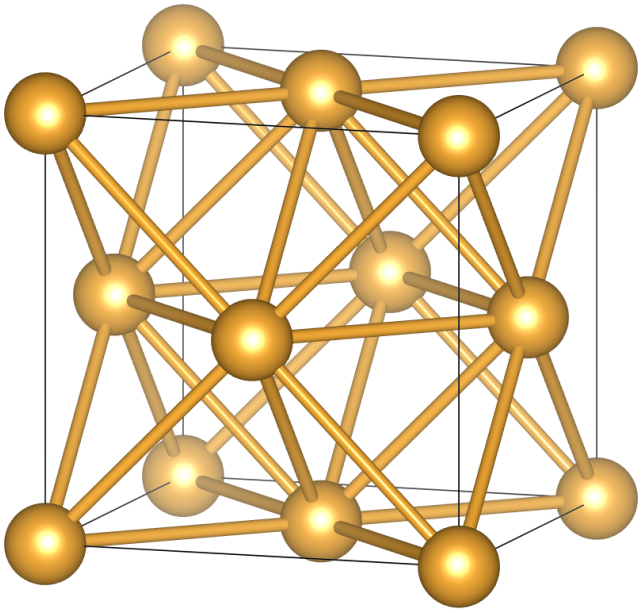

Figure 4.1: Bulk fcc gold.

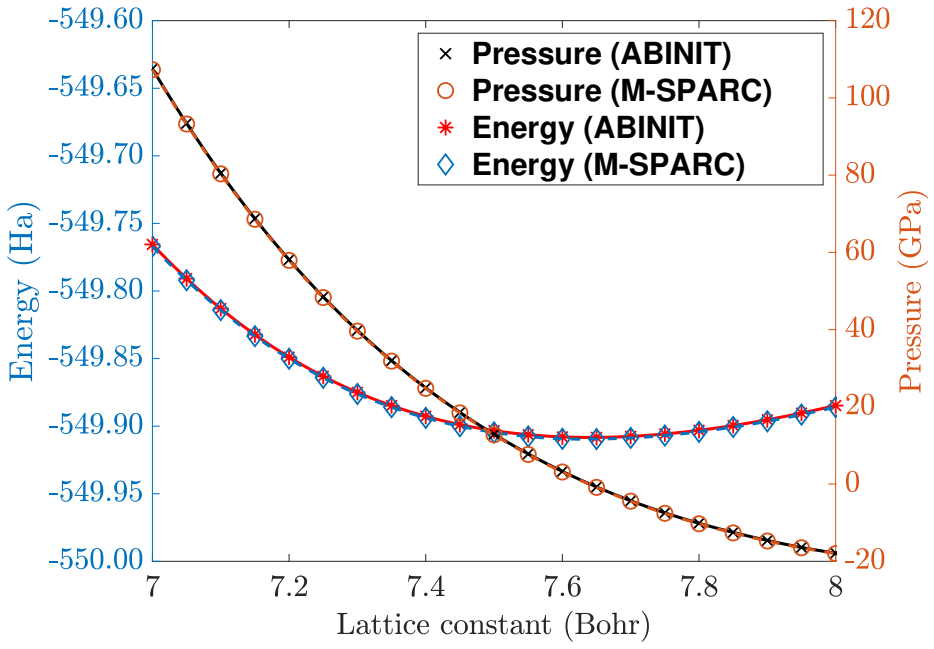

Figure 4.2: Energy and pressure and their cubic spline fit of bulk fcc gold.

Table 4.1: Lattice constant (a) and cohesive energy (E_{coh}) of bulk fcc gold.

	M-SPARC	ABINIT
a (Bohr)	7.64	7.64
E_{coh} (eV)	4.45	4.45

4.2 Germanene

In this section, we demonstrate the functionality of M-SPARC to accurately calculate two-dimensional systems, where periodic boundary conditions are applied in two dimensions and Dirichlet boundary conditions are applied in the other dimension. We choose a 49-atom cell of germanene (5×5 supercell with a vacancy, illustrated in Figure 4.3) with GGA, 2×2 grid for Brillouin zone integration, and mesh-size of 0.3 Bohr. In ABINIT, We use a plane-wave cutoff of 60 Ha. The free energies of the germanene system with and without a vacancy are shown in Table 4.2. We see that the results obtained from M-SPARC agree with those from ABINIT. Moreover, in Figure 4.4, we also visualize the perturbation of the electron density introduced by the vacancy, using the data conveniently provided by M-SPARC (the electron density $\rho(\mathbf{x})$ stored in `S.rho`). The simulation with and without vacancy takes 30 hours and 28.6 hours in M-SPARC, respectively.

Table 4.2: Free energy of germanene with and without vacancy.

E (Ha/atom)	M-SPARC	ABINIT
w/o vacancy	$-70.7682\,(-70.7682^*)$	-70.7681
w/ vacancy	$-70.7679\,(-70.7661^*)$	-70.7664

* Periodic boundary conditions

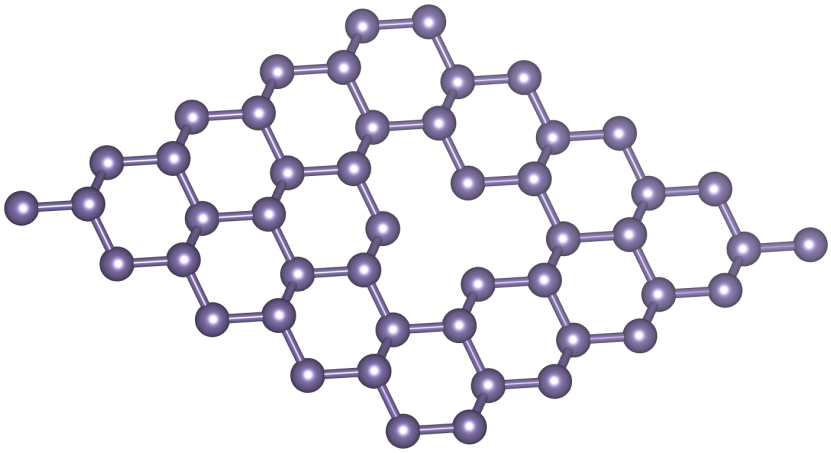

Figure 4.3: 5×5 germanene with a vacancy.

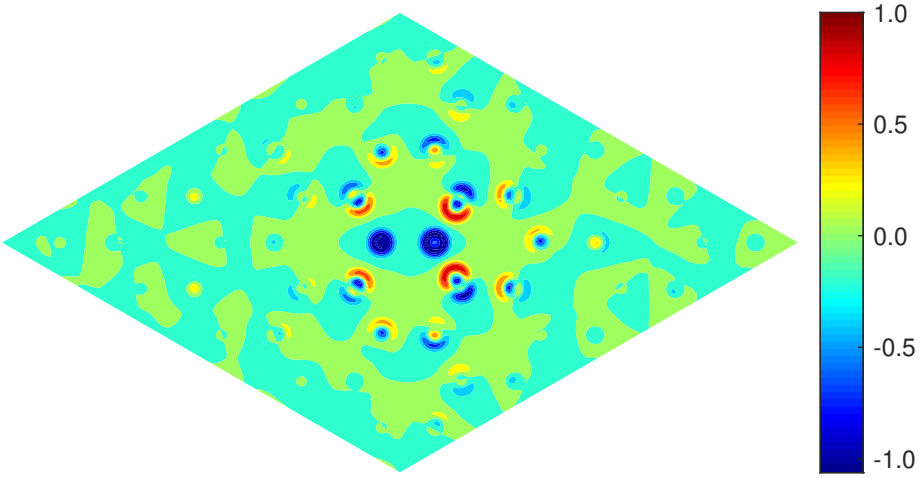

Figure 4.4: Energy density perturbation of germanene with a vacancy.

4.3 Carbon nanotube

In this section, we demonstrate M-SPARC's capability of dealing with one-dimensional systems. A typical example is the carbon nanotube, see Figure 4.5 for an illustration. We choose a 12-atom (3,3) carbon nanotube with GGA, 10 points for Brillouin zone integration, and a mesh-size of 0.3 Bohr. We apply periodic boundary conditions in one dimension and Dirichlet boundary conditions in the other two dimensions. In ABINIT, we use a plane-wave cutoff of 60 Ha, to obtain a highly accurate reference result. In Figure 4.6, we plot the band structure obtained from both M-SPARC and ABINIT. It's clear that both curves are nearly identical in Figure 4.6, demonstrating the accuracy of M-SPARC. The band structure energies and formation energies obtained from both M-SPARC and ABINIT are listed in Table 4.3, which are also in good agreement. This simulation takes 9 minutes in M-SPARC.

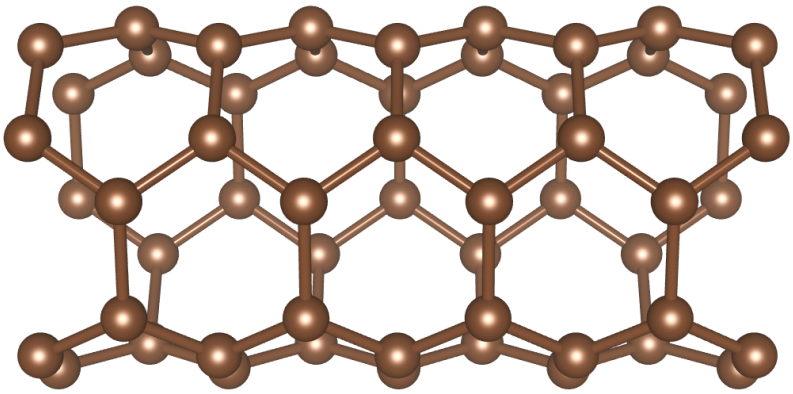

Figure 4.5: $(3, 3)$ carbon nanotube.

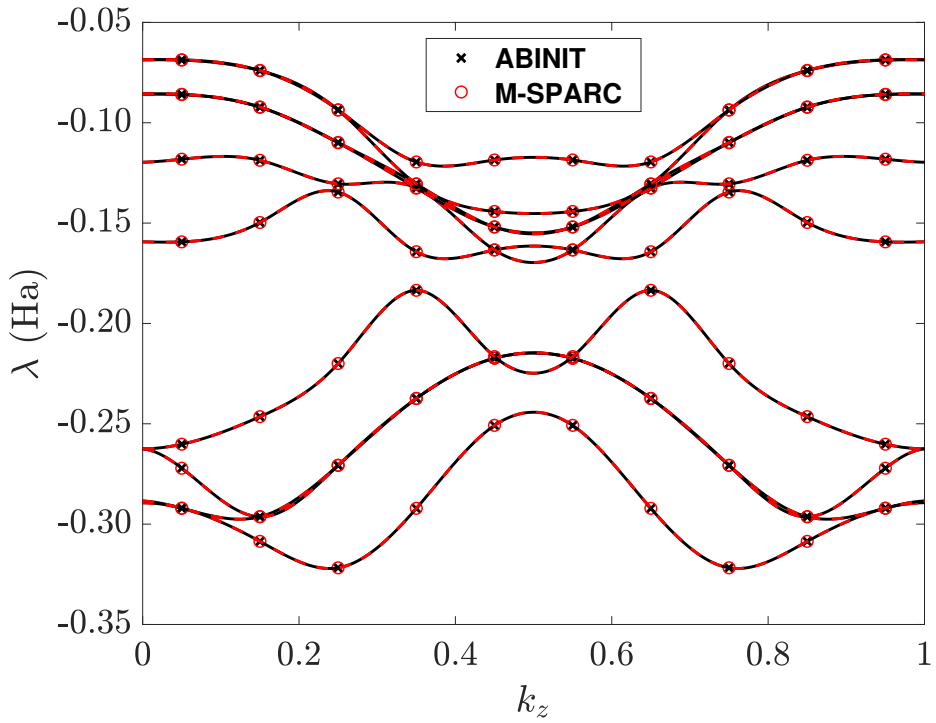

Figure 4.6: Band structure plot of $(3, 3)$ carbon nanotube.

Table 4.3: Band structure energy (E_{band}) and formation energy (E_{form}).

	M-SPARC	ABINIT
E_{band} (Ha/atom)	-1.9095	-1.9095
E_{form} (eV/atom)	1.09	1.09

4.4 Nanodot

In this section, we demonstrate the ability of M-SPARC to calculate finite systems. We choose a $Si_{275}H_{172}$ nanodot (illustrated in Figure 4.7) with LDA and mesh-size of 0.5 Bohr. We apply Dirichlet boundary conditions in all three dimensions. In ABINIT, we use a plane-wave cutoff of 60 Ha.

In Figure 4.8, we present the density of states (DOS) computed by M-SPARC and ABINIT. We also present the HOMO-LUMO gap and the free energy in Table 4.4. It's clear that there's great agreement between M-SPARC and ABINIT. This simulation takes 3 hours in M-SPARC.

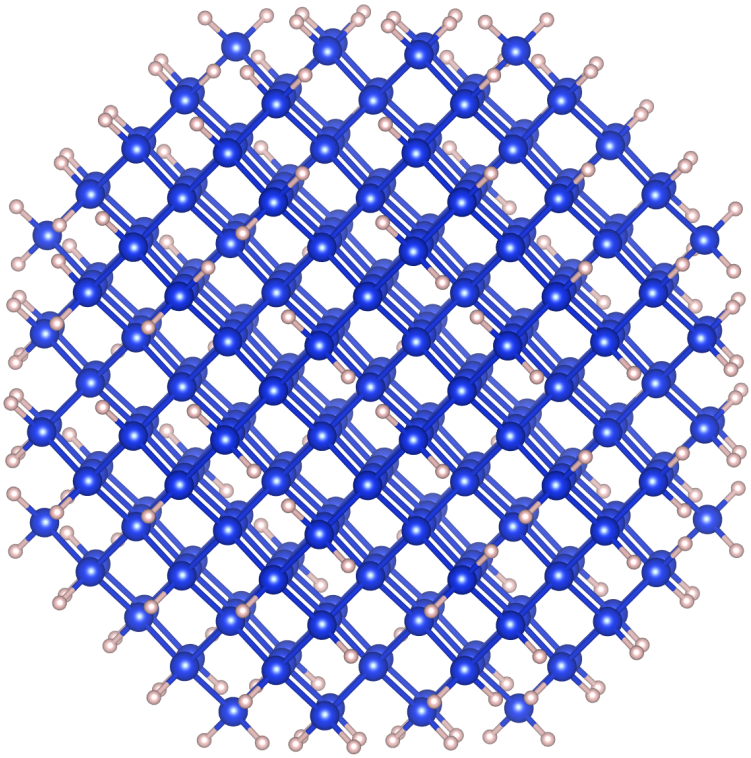

Figure 4.7: $Si_{275}H_{172}$ nanodot.

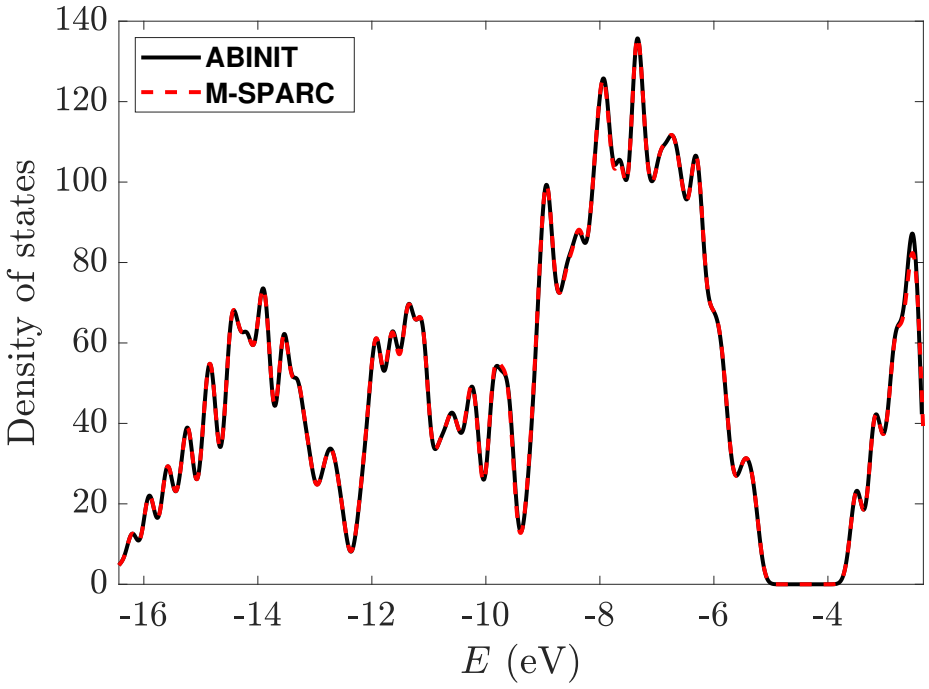

Figure 4.8: Density of states of the $Si_{275}H_{172}$ nanodot.

Table 4.4: HOMO-LUMO gap and free energy (E) of $Si_{275}H_{172}$ nanodot.

	M-SPARC	ABINIT
HOMO-LUMO gap (eV)	1.70	1.70
E (Ha/atom)	-2.6596	-2.6591

CHAPTER 5
IMPACT

M-SPARC provides a rapid prototyping platform for the development and testing of new algorithms/methods in Kohn-Sham DFT, particularly for the real-space method. This is evidenced by the number of accurate and efficient methods developed in M-SPARC that have resulted in immediate publications, including the linear scaling Spectral Quadrature (SQ) DFT method that is identically applicable to insulating and metallic systems [52]; coarse-grained DFT formulation that enables the study of crystal defects at realistic concentrations [27]; cyclic and helical symmetry-adapted DFT formulations that allow for the highly efficient study of systems with such symmetries, enabling the ab initio study of nanomaterials subjected to bending and torsional deformations [53, 54]; Kronecker product formulation of the kinetic energy operator that reduces the cost of real-space DFT for non-orthogonal systems [42]; the Discontinuous Discrete Basis Projection (DDBP) method that significantly increases the efficiency of real-space DFT by reducing the size of the Hamiltonian by up to three orders of magnitude [55]; real-space formulation for the Hellmann-Feynman stress tensor in DFT [34]; real-space formulation for isotropic Fourier-space preconditioners that can accelerate the SCF iteration in DFT [38].

M-SPARC and its variants are currently being used by multiple research groups. Moving forward, the user base is expected to significantly expand, given the current open source software release of M-SPARC and the noticeable emphasis placed on the development of real-space DFT [17, 18, 19, 20, 21, 22, 23, 24]. Possible avenues for M-SPARC to have an immediate impact in real-space DFT include: implementation of sophisticated exchange-correlation functionals such as hybrids; preconditioners for accelerating the convergence of eigensolvers; mixing schemes and associated preconditioners for accelerating SCF convergence, particularly for spin polarized calculations; techniques for reducing the size of the

Hamiltonian by projection onto a significantly smaller basis; novel boundary conditions that accurately and efficiently capture the physics/chemistry of the system; formulations for reducing the eggbox effect; and machine learning models in the context of DFT. Note that even though the main utility of M-SPARC is in rapid prototyping for large-scale implementations, it provides a convenient avenue for the ab initio investigation of small to moderately sized systems, wherein the time to solution (i.e., wall time) is only a few fold larger than established parallel DFT codes.

CHAPTER 6
CONCLUSION

This work has presented a mature MATLAB code, M-SPARC, for performing real-space Kohn-Sham DFT calculations. It can perform spin-polarized and unpolarized calculations for isolated systems such as molecules as well as extended systems such as crystals, surfaces, and nanowires. Moreover, it is capable of dealing with various boundary conditions in the treatment of electrostatics in these systems, which is sometimes impossible for plane-wave codes. Specifically, it can currently perform single-point calculations for a given atomic configuration, structural relaxations with respect to atomic positions or cell volume, and NVE molecular dynamics simulations.

Written exclusively in MATLAB language, M-SPARC provides great convenience in testing and post-processing within MATLAB. Moreover, without the complication of parallel computing, it provides a rapid prototyping platform for the development and testing of new algorithms and methods in Kohn-Sham DFT, particularly for the real-space method. M-SPARC and its variants have already resulted in several publications [52, 27, 53, 54, 55, 34, 38], and it is currently being used by multiple research groups. Given the open-source release of M-SPARC and the current noticeable emphasis placed on real-space DFT, the user base is expected to expand significantly.

As a result, M-SPARC has the potential to significantly accelerate the rate of advancements in the field, which can enable a number of new and exciting applications in science and engineering that were previously intractable.

Appendices

APPENDIX A
ELECTROSTATIC SELF ENERGY AND CORRECTION FOR PSEUDOCHARGE DENSITIES

In the local electrostatic formulation adopted in M-SPARC, the total contribution of the self energy and the repulsive correction can be written as [30, 31, 24]

$$\begin{aligned} E_{sc}(\mathbf{R}) &= \frac{1}{2}\int_{\Omega}\left(\tilde{b}(\mathbf{x};\mathbf{R})+b(\mathbf{x};\mathbf{R})\right)V_c(\mathbf{x};\mathbf{R})\,\mathrm{d}\mathbf{x} \\ &\quad -\frac{1}{2}\sum_{J}\int_{\Omega}\tilde{b}_J(\mathbf{x};\mathbf{R}_J)\tilde{V}_J(\mathbf{x};\mathbf{R}_J)\,\mathrm{d}\mathbf{x}\,, \end{aligned} \tag{A.1}$$

where

$$V_c(\mathbf{x};\mathbf{R}) = \sum_{J}\left(\tilde{V}_J(\mathbf{x};\mathbf{R}_J) - V_J(\mathbf{x};\mathbf{R}_J)\right)\,, \tag{A.2}$$

the summation index J runs over all atoms in Ω and their images, in other words, all atoms in $\mathbb{R}^3$. In addition, $\tilde{b}$ denotes the reference pseudocharge density, and $\tilde{b}_J$ represents the spherically symmetric and compactly supported reference charge density of the J^{th} atom that generates the reference potential $\tilde{V}_J$, i.e.,

$$\tilde{b}(\mathbf{x};\mathbf{R}) = \sum_{J}\tilde{b}_J(\mathbf{x};\mathbf{R}_J)\,, \qquad \tilde{b}_J(\mathbf{x};\mathbf{R}_J) = -\frac{1}{4\pi}\nabla^2\tilde{V}_J(\mathbf{x};\mathbf{R}_J)\,, \tag{A.3}$$

$$\int_{\Omega}\tilde{b}(\mathbf{x};\mathbf{R})\,\mathrm{d}\mathbf{x} = -N_e\,, \qquad \int_{\mathbb{R}^3}\tilde{b}_J(\mathbf{x};\mathbf{R}_J)\,\mathrm{d}\mathbf{x} = Z_J\,, \tag{A.4}$$

where Z_J is the valence charge of the J^{th} nucleus.

The corresponding contribution to the forces can be written as [30, 31, 24]

$$\begin{aligned} \mathbf{f}_{sc,J}(\mathbf{R}) &= -\frac{E_{sc}(\mathbf{R})}{\mathbf{R}_J} \\ &= \frac{1}{2}\sum_{J'}\int_{\Omega}\Bigg[\left(b(\mathbf{x};\mathbf{R})+\tilde{b}(\mathbf{x};\mathbf{R})\right)\left(\nabla\tilde{V}_{J'}(\mathbf{x};\mathbf{R}_{J'}) - \nabla V_{J'}(\mathbf{x};\mathbf{R}_{J'})\right) \end{aligned}$$

$$- \left(b_{J'}(\mathbf{x};\mathbf{R}_{J'}) + \tilde{b}_{J'}(\mathbf{x};\mathbf{R}_{J'}) \right) \nabla V_c(\mathbf{x};\mathbf{R}) \Big] \, \mathrm{d}\mathbf{x} \,, \tag{A.5}$$

where the summation index J' runs over all periodic images of the J^{th} atom as well as the J^{th} atom itself.

In addition, the electrostatic correction to the stress tensor components, $\sigma_{\alpha\beta}^{E_{sc}}$, can be obtained by [34]

$$\begin{aligned} \sigma_{\alpha\beta}^{E_{sc}}(\mathbf{R}) &= \frac{1}{2|\Omega|} \sum_I \int_\Omega \Big[\left(\nabla_\alpha \tilde{b}_I(\mathbf{x};\mathbf{R}_I) + \nabla_\alpha b_I(\mathbf{x};\mathbf{R}_I) \right) V_c(\mathbf{x};\mathbf{R}) - \nabla_\alpha \tilde{b}_I(\mathbf{x};\mathbf{R}_I) \tilde{V}_I(\mathbf{x};\mathbf{R}_I) \\ &- \tilde{b}_I(\mathbf{x};\mathbf{R}_I) \nabla_\alpha \tilde{V}_I(\mathbf{x};\mathbf{R}_I) + \left(\tilde{b}(\mathbf{x};\mathbf{R}) + b(\mathbf{x};\mathbf{R}) \right) \left(\nabla_\alpha \tilde{V}_I(\mathbf{x};\mathbf{R}_I) - \nabla_\alpha V_I(\mathbf{x};\mathbf{R}_I) \right) \Big] \\ &\times \left(\mathbf{x} - \mathbf{R}_I \right)_\beta \mathrm{d}\mathbf{x} + \frac{\delta_{\alpha\beta}}{|\Omega|} E_{sc}(\mathbf{R}) \,, \end{aligned} \tag{A.6}$$

where the summation index I runs over all atoms in $\mathbb{R}^3$.

APPENDIX B

EXAMPLE INPUT FILES

We demonstrate annotated input files for an example test system, the silane (SiH_4) molecule, as illustrated in Figure B.1. The input files are available in the `M-SPARC/tests/examples/` folder.

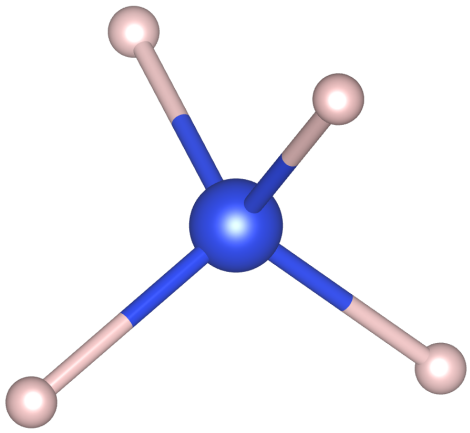

Figure B.1: Silane (SiH_4) molecule.

The first input file is the `SiH4.inpt` file, which contains user options and parameters. Refer to the user manual for a complete list of all input options, which is located in the `M-SPARC/doc/` folder.

```
CELL: 13 13 13     # Cell dimensions
FD_GRID: 26 26 26 # Finite-difference grid intervals
                   # An alternative way is to set the mesh-spacing
                   # MESH_SPACING: 0.5
BC: D D D          # Boundary conditions, 'D' represents Dirichlet
                   # B.C., 'P' represents periodic B.C.
EXCHANGE_CORRELATION: LDA_PW # LDA with the Perdew-Wang parametrization
MAXIT_SCF: 30      # maximum number of SCF iterations
TOL_SCF: 1e-5      # SCF tolerance, defined as the normalized residual
```

```
                    # of either the density or the potential.
TOL_PSEUDOCHARGE: 1e-5 # pseudocharge tolerance
NSTATES: 10         # number of states
```

The second input file is the `SiH4.ion` file, which contains the atomic informations.

```
ATOM_TYPE: Si                 # atom type
PSEUDO_POT: ../../Pseudopotentials/Si.psp8 # path to the pseudopotential
N_TYPE_ATOM: 1                # number of atoms of this type
# Cartesian coordinates follows, use 'COORD_FRAC:' to
# provide fractional coordinates.
COORD:
6.5 6.5 6.5  # Si

ATOM_TYPE: H                  # atom type
PSEUDO_POT: ../../Pseudopotentials/H.psp8 # path to the pseudopotential
N_TYPE_ATOM: 4                # number of atoms of this type
# Note: don't put any empty or commenting lines in between the coordinates
COORD:                        # coordinates follows
8.127432021000001   8.127432021000001   8.127432021000001 # H atoms
4.872567978999999   4.872567978999999   8.127432021000001
4.872567978999999   8.127432021000001   4.872567978999999
8.127432021000001   4.872567978999999   4.872567978999999
```

REFERENCES

[1] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Physical Review*, vol. 136, no. 3B, B864, 1964.

[2] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Physical Review*, vol. 140, no. 4A, A1133–A1138, 1965.

[3] A. D. Becke, "Perspective: Fifty years of density-functional theory in chemical physics," *The Journal of Chemical Physics*, vol. 140, no. 18, 18A301, 2014.

[4] A. Pribram-Jones, D. A. Gross, and K. Burke, "DFT: A theory full of holes?" *Annual Review of Physical Chemistry*, vol. 66, pp. 283–304, 2015.

[5] R. Martin, *Electronic Structure: Basic theory and practical methods*. Cambridge University Press, 2004.

[6] S. Goedecker, "Linear scaling electronic structure methods," *Reviews of Modern Physics*, vol. 71, no. 4, pp. 1085–1123, 1999.

[7] D. R. Bowler and T. Miyazaki, "O(N) methods in electronic structure calculations," *Reports on Progress in Physics*, vol. 75, no. 3, p. 036 503, 2012.

[8] P. Suryanarayana, P. P. Pratapa, A. Sharma, and J. E. Pask, "SQDFT: Spectral Quadrature method for large-scale parallel O(N) Kohn–Sham calculations at high temperature," *Computer Physics Communications*, vol. 224, pp. 288–298, 2018.

[9] P. Suryanarayana, "On nearsightedness in metallic systems for O(N) Density Functional Theory calculations: A case study on aluminum," *Chemical Physics Letters*, vol. 679, pp. 146–151, 2017.

[10] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, "KSSOLV–a MATLAB toolbox for solving the Kohn-Sham equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 2, p. 10, 2009.

[11] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Physical Review B*, vol. 54, no. 16, pp. 11 169–11 186, 1996.

[12] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, *et al.*, "Quantum espresso: A modular and open-source software project for quantum simulations of materials," *Journal of physics: Condensed matter*, vol. 21, no. 39, p. 395 502, 2009.

[13] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. I. Probert, K. Refson, and M. C. Payne, "First principles methods using castep," *Zeitschrift für Kristallographie-Crystalline Materials*, vol. 220, no. 5/6, pp. 567–570, 2005.

[14] X. Gonze, J. M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G. M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J. Y. Raty, and D. C. Allan, "First-principles computation of material properties: The ABINIT software project," *Computational Materials Science*, vol. 25, 478–492(15), 2002.

[15] F. Gygi, "Architecture of Qbox: A scalable first-principles molecular dynamics code," *IBM Journal of Research and Development*, vol. 52, no. 1.2, pp. 137–144, 2008.

[16] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, *et al.*, "NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations," *Computer Physics Communications*, vol. 181, no. 9, pp. 1477–1489, 2010.

[17] J Bernholc, J.-Y. Yi, and D. Sullivan, "Structural transitions in metal clusters," *Faraday Discussions*, vol. 92, pp. 217–228, 1991.

[18] J. R. Chelikowsky, N. Troullier, and Y. Saad, "Finite-difference-pseudopotential method: Electronic structure calculations without a basis," *Physical Review Letters*, vol. 72, no. 8, p. 1240, 1994.

[19] A. P. Seitsonen, M. J. Puska, and R. M. Nieminen, "Real-space electronic-structure calculations: Combination of the finite-difference and conjugate-gradient methods," *Physical Review B*, vol. 51, no. 20, p. 14 057, 1995.

[20] F. Gygi and G. Galli, "Real-space adaptive-coordinate electronic-structure calculations," *Physical Review B*, vol. 52, no. 4, R2229, 1995.

[21] E. L. Briggs, D. J. Sullivan, and J. Bernholc, "Real-space multigrid-based approach to large-scale electronic structure calculations," *Phys. Rev. B*, vol. 54, pp. 14 362–14 375, 20 1996.

[22] J.-I. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi, S. Okada, and K. Yabana, "A massively-parallel electronic-structure calculations based on real-space density functional theory," *Journal of Computational Physics*, vol. 229, no. 6, pp. 2339–2363, 2010.

[23] S. Ghosh and P. Suryanarayana, "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Isolated clusters," *Computer Physics Communications*, vol. 212, pp. 189–204, 2017.

[24] S. Ghosh and P. Suryanarayana, "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Extended systems," *Computer Physics Communications*, vol. 216, pp. 109–125, 2017.

[25] Y. Hasegawa, J.-I. Iwata, M. Tsuji, D. Takahashi, A. Oshiyama, K. Minami, T. Boku, F. Shoji, A. Uno, M. Kurokawa, *et al.*, "First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the k computer," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, 2011, p. 1.

[26] A. Natan, A. Benjamini, D. Naveh, L. Kronik, M. L. Tiago, S. P. Beckman, and J. R. Chelikowsky, "Real-space pseudopotential method for first principles calculations of general periodic and partially periodic systems," *Physical Review B*, vol. 78, no. 7, p. 075 109, 2008.

[27] P. Suryanarayana, K. Bhattacharya, and M. Ortiz, "Coarse-graining Kohn-Sham density functional theory," *Journal of the Mechanics and Physics of Solids*, vol. 61, no. 1, pp. 38–60, 2013.

[28] S. Ghosh, A. S. Banerjee, and P. Suryanarayana, "Symmetry-adapted real-space density functional theory for cylindrical geometries: Application to large group-IV nanotubes," *Physical Review B*, vol. 100, no. 12, p. 125 143, 2019.

[29] D. Osei-Kuffuor and J.-L. Fattebert, "Accurate and scalable O(N) algorithm for first-principles molecular-dynamics computations on large parallel computers," *Physical Review Letters*, vol. 112, no. 4, p. 046 401, 2014.

[30] P. Suryanarayana and D. Phanish, "Augmented Lagrangian formulation of orbital-free density functional theory," *Journal of Computational Physics*, vol. 275, no. 0, pp. 524–538, 2014.

[31] S. Ghosh and P. Suryanarayana, "Higher-order finite-difference formulation of periodic orbital-free density functional theory," *Journal of Computational Physics*, vol. 307, pp. 634–652, 2016.

[32] W. RussellBurdick, Y. Saad, L. Kronik, I. Vasiliev, M. Jain, and J. R. Chelikowsky, "Parallel implementation of time-dependent density functional theory," *Computer Physics Communications*, vol. 156, no. 1, pp. 22–42, 2003.

[33] L. Kleinman and D. M. Bylander, "Efficacious form for model pseudopotentials," *Physical Review Letters*, vol. 48, no. 20, p. 1425, 1982.

[34] A. Sharma and P. Suryanarayana, "On the calculation of the stress tensor in real-space kohn-sham density functional theory," *The Journal of Chemical Physics*, vol. 149, no. 19, p. 194 104, 2018.

[35] D. Alfe, "Ab initio molecular dynamics, a simple algorithm for charge extrapolation," *Computer Physics Communications*, vol. 118, no. 1, pp. 31–33, 1999.

[36] P. P. Pratapa and P. Suryanarayana, "Restarted Pulay mixing for efficient and robust acceleration of fixed-point iterations," *Chemical Physics Letters*, vol. 635, pp. 69–74, 2015.

[37] A. S. Banerjee, P. Suryanarayana, and J. E. Pask, "Periodic Pulay method for robust and efficient convergence acceleration of self-consistent field iterations," *Chemical Physics Letters*, vol. 647, pp. 31–35, 2016.

[38] S. Kumar, Q. Xu, and P. Suryanarayana, "On preconditioning the self-consistent field iteration in real-space density functional theory," *Chemical Physics Letters*, vol. 739, p. 136 983, 2020.

[39] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, "Self-consistent-field calculations using Chebyshev-filtered subspace iteration," *Journal of Computational Physics*, vol. 219, no. 1, pp. 172–184, 2006.

[40] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, "Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration," *Physical Review E*, vol. 74, no. 6, p. 066 704, 2006.

[41] Y. Zhou, J. R. Chelikowsky, and Y. Saad, "Chebyshev-filtered subspace iteration method free of sparse diagonalization for solving the Kohn–Sham equation," *Journal of Computational Physics*, vol. 274, pp. 770–782, 2014.

[42] A. Sharma and P. Suryanarayana, "On real-space density functional theory for non-orthogonal crystal systems: Kronecker product formulation of the kinetic energy operator," *Chemical Physics Letters*, vol. 700, pp. 156–162, 2018.

[43] P. P. Pratapa, P. Suryanarayana, and J. E. Pask, "Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems," *Journal of Computational Physics*, vol. 306, pp. 43–54, 2016.

[44] P. Suryanarayana, P. P. Pratapa, and J. E. Pask, "Alternating Anderson–Richardson method: An efficient alternative to preconditioned Krylov methods for large, sparse linear systems," *Computer Physics Communications*, vol. 234, pp. 278–285, 2019.

[45] D. R. Hamann, "Optimized norm-conserving Vanderbilt pseudopotentials," *Physical Review B*, vol. 88, no. 8, p. 085 117, 2013.

[46] N. Troullier and J. L. Martins, "Efficient pseudopotentials for plane-wave calculations," *Physical Review B*, vol. 43, no. 3, pp. 1993–2006, 1991.

[47] J. P. Perdew and A. Zunger, "Self-interaction correction to density-functional approximations for many-electron systems," *Phyical Review B*, vol. 23, pp. 5048–5079, 10 1981.

[48] J. P. Perdew and W. Yue, "Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation," *Physical Review B*, vol. 33, no. 12, p. 8800, 1986.

[49] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Physical Review Letters*, vol. 77, no. 18, p. 3865, 1996.

[50] Q. Xu, A. Sharma, B. Comer, H. Huang, E. Chow, A. J. Medford, J. E. Pask, and P. Suryanarayana, "SPARC: Simulation package for ab-initio real-space calculations," *arXiv preprint arXiv:2005.10431*, 2020.

[51] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface*. MIT press, 1999, vol. 1.

[52] P. P. Pratapa, P. Suryanarayana, and J. E. Pask, "Spectral quadrature method for accurate O(N) electronic structure calculations of metals and insulators," *Computer Physics Communications*, vol. 200, pp. 96–107, 2015.

[53] A. S. Banerjee and P. Suryanarayana, "Cyclic density functional theory: A route to the first principles simulation of bending in nanostructures," *Journal of the Mechanics and Physics of Solids*, vol. 96, pp. 605–631, 2016.

[54] A. S. Banerjee and P. Suryanarayana, "Ab initio framework for simulating systems with helical symmetry: Formulation, implementation and applications to torsional deformations in nanostructures," *preprint*, 2018.

[55] Q. Xu, P. Suryanarayana, and J. E. Pask, "Discrete discontinuous basis projection method for large-scale electronic structure calculations," *The Journal of Chemical Physics*, vol. 149, no. 9, p. 094 104, 2018.